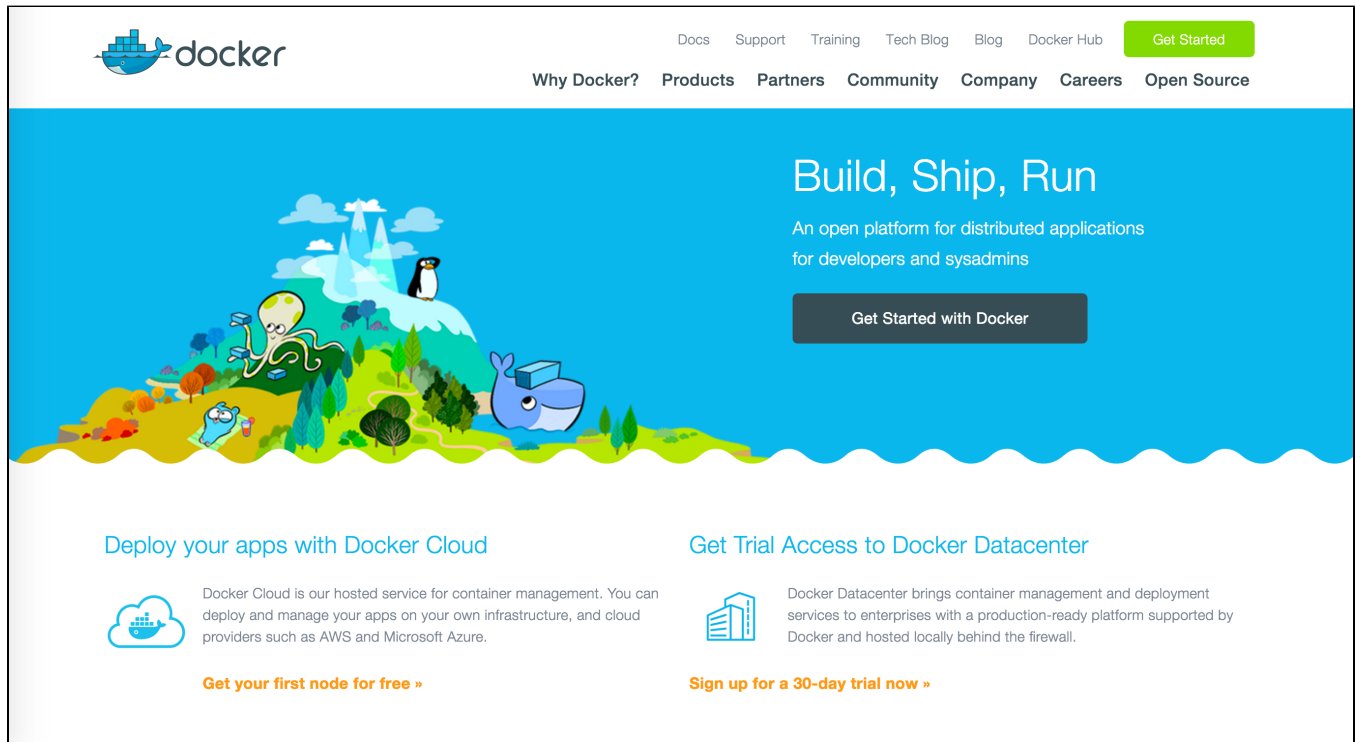


Joget on Docker

Docker (www.docker.com) is an open platform making it easier to create, deploy, and run applications by using containers.



This article describes the steps required to run Joget on Docker. With Docker, you can easily deploy Joget with just a single command. Public Joget Docker images are hosted in the Docker Hub repository (<https://hub.docker.com/u/jogetworkflow/>)

Prerequisite: Install Docker

Install Docker following the docs at <https://docs.docker.com/engine/installation/> There are instructions for Linux, Windows or Mac.

Option 1: Run Joget with Embedded MySQL

The easiest way is to run a Joget container which also contains a MySQL database:

```
docker run -d -p 8080:8080 -v /var/lib/mysql --name joget jogetworkflow/joget-enterprise
```

With just a single command, you will have an entire running installation of Joget. Browse to the installation at http://your_docker_host:8080/jw

Option 2: Run Joget with Separate MySQL and Data Volume

For better flexibility and manageability, you can run a Joget container separately from the database and shared data volume:

```
# create a volume container for shared data
docker volume create jogetdata

# run a MySQL database container
docker run -d --name jogetdb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=jwdb -e MYSQL_USER=joget -e
MYSQL_PASSWORD=joget -e MYSQL_DATABASE=jwdb mysql:5.7

# run a Joget container.
docker run -d --link jogetdb:jwdb --name joget -p 8080:8080 -e MYSQL_HOST=jwdb -e MYSQL_DATABASE=jwdb -e
MYSQL_PORT=3306 -e MYSQL_USER=joget -e MYSQL_PASSWORD=joget --mount source=jogetdata,target=/opt/joget/wflow
jogetworkflow/joget-enterprise
```

** do note that without specifying any tag, jogetworkflow/joget-enterprise will default to the "latest" tag so it is a good practice to fix the tag to your preferred version.

Browse to the installation at http://your_docker_host:8080/jw

Option 3: Run Joget using Docker Compose

You can run Joget using the docker-compose yaml template file below:

`docker-compose.yml`

```
version: '2'

services:
  joget:
    container_name: jogetapp
    image: jogetworkflow/joget-enterprise:latest
    restart: unless-stopped
    environment:
      - MYSQL_HOST=jogetdb
      - MYSQL_DATABASE=jwdb
      - MYSQL_PORT=3306
      - MYSQL_USER=joget
      - MYSQL_PASSWORD=joget
    volumes:
      - jogetdata:/opt/joget/wflow
    networks:
      joget-backend:
    ports:
      - 8080:8080
    depends_on:
      jogetdb:
        condition: service_healthy
  jogetdb:
    container_name: jogetdb
    image: mysql:8.0
    restart: unless-stopped
    environment:
      - MYSQL_ROOT_PASSWORD=rootpass
      - MYSQL_DATABASE=jwdb
      - MYSQL_USER=joget
      - MYSQL_PASSWORD=joget
    volumes:
      - mysqldata:/var/lib/mysql
    networks:
      joget-backend:
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      timeout: 20s
      retries: 10

volumes:
  jogetdata:
  mysqldata:

networks:
  joget-backend:
```

Either use `docker compose up -d` command if you are using `docker-compose.yml` filename or `docker compose -f {yourcomposefilename} up -d` if you set a custom filename for your `yml` file. You can then browse to the installation at http://your_docker_host:8080/jw

Preserve MAC Address for Joget License

Once the docker instance is up, you may want to obtain the MAC address of the docker instance by using the following command.

```
sudo docker inspect --format='{{range .NetworkSettings.Networks}}{{.MacAddress}}{{end}}' joget
```

After you obtain the MAC address, you may preserve the MAC address the next time you create the same Joget instance again to ensure that Joget license tied to it continue to be valid.

```
docker run -ti --mac-address 00:00:00:00:00:11 -d -p 8080:8080 -v /var/lib/mysql --name joget3 jogetworkflow/joget-enterprise
```